In the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Original) A method for universal programming language conversion between two different sequential programming languages including a source program in a first programming language and a target program in a second programming language, the method comprising the steps of:

parsing the source program in the first programming language using a parsing interface specific to the first programming language;

stripping all syntax from the parsed source program;

receiving as input the parsed source program without any syntax;

instantiating classes in a framework for capturing semantics of the parsed source program independent of syntax and execution model of the sequential programming languages;

producing a semantic representation of the parsed source program without any syntax; and

receiving the semantic representation at a printer interface specific to the second programming language and adding the syntax of the target program in the second programming language.

2. (Original) The method in accordance with claim 1, wherein the source program is a high level programming language and the target program is a high level programming language.

3. (Original) The method in accordance with claim 1, wherein the source program is a high level programming language and the target program is a low level programming language.

4. (Original) The method in accordance with claim 1, wherein the classes are C++ classes representing fundamental core constructs of all sequential programming languages.

5. (Original) An apparatus for universal programming language conversion between two different sequential programming languages including a source program in a first programming language and a target program in a second programming language, comprising:

a parsing interface specific to the first programming language for parsing the source program in the first programming language and stripping all syntax from the parsed source program;

a framework including instantiable classes for capturing a semantic representation of the parsed source program independent of syntax and execution model of the sequential programming languages; and

a printer interface specific to the second programming language for receiving the semantic representation and adding the syntax of the target program in the second programming language.

6. (Original) The apparatus in accordance with claim 5, wherein the source program is a high level programming language and the target program is a high level programming language.

7. (Original) The apparatus in accordance with claim 5, wherein the source program is a high level programming language and the target program is a low level programming language.

8. (Original) The apparatus in accordance with claim 5, wherein the classes are C++ classes representing fundamental core constructs of all sequential programming languages.

9. (Original) An apparatus for universal programming language conversion between two different sequential programming languages including a source program in a first programming language and a target program in a second programming language, comprising:

a processor for instantiating classes in a framework representing a unification of semantics of the sequential programming languages independent of syntax and execution model.

10. (Original) The apparatus in accordance with claim 9, wherein the sequential programming

languages are sequential procedural and sequential object oriented programming languages.

11. (New) The method in accordance with claim 1, wherein the first and second programming languages are object oriented programming languages.

12. (New) The method in accordance with claim 1, wherein the first and second programming languages are procedural programming languages.

13. (New) The apparatus in accordance with claim 5, wherein the first and second programming languages are object oriented programming languages.

14. (New) The apparatus in accordance with claim 5, wherein the first and second programming languages are procedural programming languages.

15. (New) The apparatus in accordance with claim 9, wherein the first and second programming languages are object oriented programming languages.

16. (New) The apparatus in accordance with claim 9, wherein the first and second programming languages are procedural programming languages.